

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-194899

(43) 公開日 平成11年(1999) 7月21日

(51) Int.Cl. <sup>6</sup>	識別記号	F I
G 0 6 F 3/06	5 4 0 3 0 5	C 0 6 F 3/06 5 4 0 3 0 5 C
G 1 1 B 20/10 20/18	5 7 0	G 1 1 B 20/10 A 20/18 5 7 0 Z

審査請求 未請求 請求項の数15 F D (全 16 頁)

(21) 出願番号 特願平9-366782

(22) 出願日 平成9年(1997)12月26日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 富田 治男

東京都青梅市末広町2丁目9番地 株式会  
社東芝青梅工場内

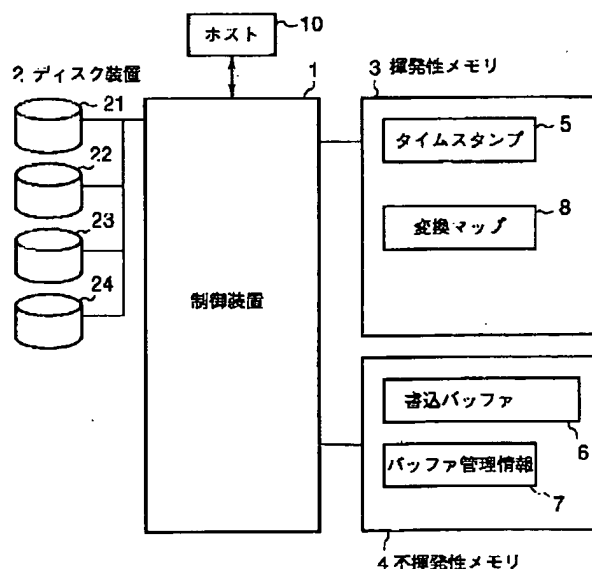
(74) 代理人 弁理士 鈴江 武彦 (外6名)

(54) 【発明の名称】 ディスク記憶システム及び同システムに適用するデータ更新方法

(57) 【要約】

【課題】高速の書き込み方法を採用したRAID方式のディスク記憶システムに適用して、特にデータ更新処理に関する性能の向上を図ることにある。

【解決手段】RAID構成のディスク記憶システムであって、ストライプに相当する記憶容量を有する書き込みバッファ6と、バッファ管理テーブル7と、制御装置1とを有し、制御装置1は書き込みバッファ6に必要に応じてデータ長を変化した論理ブロックを蓄積し、データ更新処理時に書き込みバッファ6に蓄積した論理ブロックが $N \times K - 1$ 個に達するまでその論理ブロックの更新を遅延させ、 $N \times K - 1$ 個の論理ブロックに論理アドレス・タグ・ブロックを加えた $N \times K$ 個の論理ブロックを、旧データを保持している領域とは別の空領域の中から連続した記憶領域を選択して、連続した書き込み操作により順次書き込む。



## 【特許請求の範囲】

【請求項1】 N台のディスクドライブから構成されるディスク記憶システムであって、  
 $N \times K$  ( $K$ はブロック数を意味する整数)個の論理ブロックに相当する記憶容量を有する書込みバッファ手段と、  
前記書込みバッファを前記各ディスクドライブとの間で転送される読出しデータ及び書込みデータのキャッシュメモリとして管理し、前記書込みバッファに必要な応じてデータ長を可変した前記論理ブロックを蓄積するための管理手段と、  
データ更新処理時に前記書込みバッファに蓄積した論理ブロックが $N \times K - 1$ 個に達するまでその論理ブロックの更新を遅延させ、当該 $N \times K$ 個の論理ブロックを、前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域を選択して、連続した書込み操作により順次書込むための書込み制御手段とを具備したことを特徴とするディスク記憶システム。

【請求項2】 前記管理手段は前記書込みバッファ手段と共にメモリ上に構成されたバッファ管理テーブルを有し、書込みデータをデータ長に応じて最適なブロック単位に分割し、前記書込みバッファ手段にログ形式で順番に詰めて格納し、ホストシステムからの書込みデータの論理アドレスを前記書込みバッファ手段の格納領域に対応する前記バッファ管理テーブルのエントリに格納するように構成されたことを特徴とする請求項1記載のディスク記憶システム。

【請求項3】 前記書込み制御手段は、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレス・タグ・ブロックを生成し、 $N \times K - 1$ 個の論理ブロックに前記論理アドレス・タグ・ブロックを加えた $N \times K$ 個の論理ブロックを前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域に書込みむように構成されたことを特徴とする請求項1記載のディスク記憶システム。

【請求項4】 N台のディスクドライブから構成されるディスク記憶システムに適用するデータ更新方法であって、  
 $N \times K$  ( $K$ はブロック数を意味する整数)個の論理ブロックに相当する記憶容量を有する書込みバッファ手段に前記論理ブロックのデータ長を必要に応じて可変長で可変した前記論理ブロックを蓄積するステップと、  
データ更新時に前記書込みバッファ手段に蓄積した前記論理ブロックが $N \times K - 1$ 個に達するまで、当該論理ブロックの更新を遅延させるステップと、  
当該 $N \times K$ 個の論理ブロックを、前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域を選択して、連続した

書込み操作により順次書込むステップとからなる処理を実行することを特徴とするデータ更新方法。

【請求項5】 N台のディスクドライブから構成されるディスク記憶システムに適用するデータ更新方法であって、  
 $N \times K$  ( $K$ はブロック数を意味する整数)個の論理ブロックに相当する記憶容量を有する書込みバッファ手段に前記論理ブロックのデータ長を必要に応じて可変長で可変した前記論理ブロックを蓄積するステップと、  
データ更新時に前記書込みバッファ手段に蓄積した前記論理ブロックが $N \times K - 1$ 個に達するまで、当該論理ブロックの更新を遅延させるステップと、  
前記書込みバッファ手段に蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレス・タグ・ブロックを生成するステップと、  
当該 $N \times K - 1$ 個の論理ブロックに前記論理アドレス・タグ・ブロックを加えた $N \times K$ 個の論理ブロックを生成するステップと、  
前記論理アドレス・タグ・ブロックを加えた $N \times K$ 個の論理ブロックを前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域を選択して、連続した書込み操作により順次書込むステップとからなる処理を実行することを特徴とするデータ更新方法。

【請求項6】 N台のディスクドライブから構成されるディスク記憶システムに適用するデータ更新方法であって、  
 $N \times K$  ( $K$ はブロック数を意味する整数)個の論理ブロックに相当する記憶容量を有する書込みバッファ手段に前記論理ブロックのデータ長を必要に応じて可変長で可変した前記論理ブロックを蓄積するステップと、  
データ更新時に前記書込みバッファ手段に蓄積した前記論理ブロックが $N \times K - 1$ 個に達するまで、当該論理ブロックの更新を遅延させるステップと、  
前記書込みバッファ手段に蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレス・タグ・ブロックを生成するステップと、  
前記 $N \times K - 1$ 個の論理ブロックに前記論理アドレス・タグ・ブロックを加えた $N \times K$ 個の論理ブロックから $K$ 個のバリティブロックを生成するステップと、  
当該バリティブロックを加えた $N \times K$ 個の論理ブロックを生成するステップと、  
前記バリティブロックを加えた $N \times K$ 個の論理ブロックを前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域を選択して、連続した書込み操作により順次書込むステップとからなる処理を実行することを特徴とするデータ更新方法。

【請求項7】 前記書込みバッファ手段に対する書込み操作を行なう場合には、論理アドレスに対応してディス

クドライブ上の連続したセクタの位置に対応するように論理ブロックを再配置するステップを含むことを特徴とする請求項4、請求項5、請求項6のいずれか記載のデータ更新方法。

【請求項8】 前記N×K個の論理ブロックをストライプと定義した場合に、前記ディスク装置上に連続した論理ブロックを連続して書込めるような空領域を作るため、前記複数のストライプを讀出して最新の論理ブロックだけを前記書込みバッファ手段に移して、対応する論理アドレス・タグ・ブロック内の論理アドレスから新しい論理アドレス・タグ・ブロックを生成するステップと、前記書込みバッファ手段の有効データと生成された論理アドレス・タグ・ブロックとから構成されるストライプを、讀出したストライプを保持していた領域とは別の空領域に依存するストライプの物理アドレスに順次書込むステップとを含む請求項4、請求項5、請求項6のいずれか記載のデータ更新方法。

【請求項9】 前記空領域をつくるために、データの讀出し要求や書込み要求とは独立した動作時に、讀出した前記ストライプを保持していた領域とは別の空領域のデータの配置を変更するステップを含むことを特徴とする請求項4、請求項5、請求項6のいずれか記載のデータ更新方法。

【請求項10】 ホストシステムからの論理アドレスと前記ディスクドライブの物理アドレスとを変換するめたの変換マップを設けて、アドレス変換時に前記変換マップの全てのエントリを検索するステップを含むことを特徴とする請求項4、請求項5、請求項6のいずれか記載のデータ更新方法。

【請求項11】 前記アドレス変換時に前記変換マップから論理アドレスをハッシュキーにした線形検索を実行するステップを含むことを特徴とする請求項10記載のデータ更新方法。

【請求項12】 前記論理アドレスから前記物理アドレスへのアドレス変換時に、前記変換マップを木構造で管理し、二分木検索を実行するステップを含むことを特徴とする請求項10記載のデータ更新方法。

【請求項13】 前記各論理アドレスに対するストライプ番号、ストライプ内のブロック番号、有効データのタイムスタンプから構成される前記変換マップを生成し、前記論理アドレス・タグ・ブロックに基づいて変換マップの一部を前記ディスクドライブ内の空領域に格納するステップを含むことを特徴とする請求項10記載のデータ更新方法。

【請求項14】 前記変換マップの一部を前記ディスクドライブに格納する場合に、最も最後にアクセスされたデータを含む論理ブロックに対応する領域に応じて順次格納するステップを含むことを特徴とする請求項13記載のデータ更新方法。

【請求項15】 前記書込み制御手段は、前記ディスクドライブ上のセクタ位置に対応して、前記ディスクドライブ上の更新対象の旧データを保持している領域を選択して記憶領域に書込むことを特徴とする請求項4、請求項5、請求項6のいずれか記載のデータ更新方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、特にRAID (Redundant Array of Inexpensive Disk) と呼ぶディスクアレイ装置から構成されるディスク記憶システムに適用し、具体的にはデータ更新機能を改善したディスク記憶システムに関する。

【0002】

【従来の技術】従来、複数のディスクドライブから構成されるディスクアレイ装置 (RAID と呼ぶ) が周知である。RAID は、例えばネットワーク・サーバに接続されて、大容量のデータを記憶できるディスク記憶システムである。このようなRAID に関して、特に高速のデータ書込み方法として、データ更新処理時に更新データを旧データの記憶領域に書換えることなく、更新データを一括して書込む方法が提案されている。具体的には、更新データを用意した書込みバッファに蓄積し、この蓄積した更新データをディスクドライブ内に予め用意した空領域に一括して書込む方法である (特開平6-214720号公報、特開平6-266510号公報、特願平9-214645号公報を参照)。

【0003】図22を参照して、従来の書込み方法 (データ更新方法) について簡単に説明する。図22において、L1～L99は論理ブロックを意味し、P1～P56はディスク装置 (ディスクドライブ) 内の物理ブロックを意味する。ここでは、論理ブロックL6、L4、L2、L12、L7、L11のそれぞれを更新する場合を想定する。これらの論理ブロックの各旧データは、ディスク装置内の物理ブロックP6、P4、P2、P12、P7、P11に存在する。従って、通常の手書き込み方法では、これらの物理ブロックP6、P4、P2、P12、P7、P11の内容を更新することになる。しかし、高速の手書き込み方法では、物理ブロックP6、P4、P2、P12、P7、P11の各データはそのまま維持して、新しい論理ブロックL6、L4、L2、L12、L7、L11の各データを予め用意した別の空領域である物理ブロックP51、P52、P53、P54、P55、P56に一括して書込む。これにより、通常の手書き込み方法では6回の書込み操作が必要であるが、高速の手書き込み方法では例えば2物理ブロックを一括して書込む場合には3回の書込み操作に減少させることが可能であり、書込み性能を向上させることができる。

【0004】ここで、図22に示す具体例では、便宜的に1ディスクに2物理ブロックを一括して書込む場合を示しているが、実際には数十物理ブロックが一括して書

込まれる。また、レベル4のRAID方式(RAID4方式)およびレベル5のRAID方式(RAID5方式)では、1個のストライプ( $N \times K$ 個の論理ブロックであり、 $N$ はドライブ数、 $K$ はブロック数)を一括して書換えることが可能であるため、パリティ維持のためのディスク読出し動作も不要であり、書込み時のオーバーヘッドを減少させることも可能である。なお、RAID4方式は、データの配置単位をビットやバイトのような小さな単位ではなく、セクタやブロックのような大きな単位とし、小容量のデータ読出し要求に対してディスクを独立動作可とする方式である。また、RAID5方式は、冗長データ(パリティデータ)を専用のパリティディスクに格納するのではなく、各データディスクに巡回的に配置する方式である。

【0005】一方、論理ブロック $L_6$ 、 $L_5$ 、 $L_2$ 、 $L_{12}$ 、 $L_7$ 、 $L_{11}$ の各最新データはディスク装置内の物理ブロック $P_{51} \sim P_{56}$ に存在するので、間接マップの内容を正しいディスク位置を指示するように書き換える。また、論理ブロックのデータを読出すときには、この間接マップを参照して、最新のディスク位置を求めてから読出すので、旧データを読出すような不都合は発生しない。

【0006】

【発明が解決しようとする課題】前記のような従来の高速の書込み方法では、図22に示すように、不揮発性メモリ上に書込みバッファを用意し、この書込みバッファを介してディスク装置内の空領域をブロック単位に分割し、空ブロックから順番に詰めて格納している。このため、読出すデータが連続である場合にもブロック単位に分割してから、ブロック毎のアドレスに変換する必要がある。従って、ホストシステムから論理ブロックのデータ長を越える読込み要求があった場合に、当該要求データは複数の論理ブロックに分割されるという問題点がある。

【0007】また、特にデータ更新処理においては、毎回更新すべきデータがバッファ管理テーブルに対応するエントリに存在するかどうかを変換マップを使用して検索する必要があるため、この検索に要する分だけ処理時間が増大する問題がある。さらに、変換マップはディスク装置の記憶容量に依存して、システムのメインメモリ上に作成する必要があるため、当該メインメモリの使用容量を制限する要因になっている。

【0008】そこで、本発明の目的は、特にRAID方式のディスク記憶システムにおいて、高速の書込み方法を適用した場合のデータ更新処理に関する性能の向上を図ることにある。

【0009】

【課題を解決するための手段】本発明は、特にRAID方式のディスク記憶システムにおいて、連続したデータの読出し要求に応じた読出し処理時に、原理的に論理ブ

ロックを分割することなく読出し処理を実行する方式である。また、データ更新処理時にバッファ管理テーブルに対する変換マップのエントリを検索するための検索処理の効率化を実現する方式である。

【0010】具体的には、本発明は、 $N$ 台のディスクドライブから構成されるディスク記憶システムであって、 $N \times K$  ( $K$ はブロック数を意味する整数)個の論理ブロックに相当する記憶容量を有する書込みバッファ手段と、前記書込みバッファを前記各ディスクドライブとの間で転送される読出しデータ及び書込みデータのキャッシュメモリとして管理し、前記書込みバッファに必要に応じてデータ長を変化した前記論理ブロックを蓄積するキャッシュ管理手段と、データ更新処理時に前記書込みバッファに蓄積した論理ブロックが $N \times K - 1$ 個に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレス・タグ・ブロックを生成し、 $N \times K - 1$ 個の論理ブロックに前記論理アドレス・タグ・ブロックを加えた $N \times K$ 個の論理ブロックを、前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域(セクタ単位)を選択して、連続した書込み操作により順次書込むためのデータ更新手段とを有するシステムである。

【0011】また、本発明の別の観点として、前記データ更新手段は、データ更新処理時に前記書込みバッファに蓄積した論理ブロックが $N \times K - 1$ 個に達するまでその論理ブロックの更新を遅延させ、前記書込みバッファに蓄積された各論理ブロックに対する論理アドレスから構成される論理アドレス・タグ・ブロックを生成し、 $N \times K - 1$ 個の論理ブロックに前記論理アドレス・タグ・ブロックを加えた $N \times K$ 個の論理ブロックから $K$ 個のパリティブロックを生成し、この論理ブロックにパリティブロックを加えた $N \times K$ 個の論理ブロックを、前記各ディスクドライブ上の更新対象の旧データを保持している領域とは別の空領域の中から連続した記憶領域(セクタ単位)を選択して、連続した書込み操作により順次書込む機能を有するように構成されたディスク記憶システムである。

【0012】このような構成により、読出すデータが連続である場合には、不揮発性メモリ上に設ける書込みバッファの空領域を論理ブロック単位に分割する必要性を無くすことが可能となる。

【0013】

【発明の実施の形態】以下図面を参照して本発明の実施の形態を説明する。

【0014】図1及び図23は本実施形態のRAID方式のディスク記憶システムの概念的構成を示すブロック図である。なお、本発明に関する記述の全体を通じて、より詳細に説明するために図1に示す実施形態を中心と

して説明するが、当該技術分野に属する熟練者であれば、図1及び図23から派生するシステム構成であって、本発明を実施することが可能であればよい。例えば、図1において、ホストシステム10と制御装置1とのインターフェースは、SCSIインターフェースであっても良いし、PCIインターフェースであってもよい。なお、図23のシステムは、図1に示すバッファ管理テーブル7に相当するバッファ管理情報23-7を揮発性メモリ3である主記憶メモリ23-3に設けた場合であり、ほかの構成要素は図1に示すシステムと同様である。

(システム構成) 本システムは、図1に示すように、制御装置(CPU)1、ディスク装置2(ディスクドライブ21, 22, 23, 24)、揮発性メモリ3、および不揮発性メモリ4から構成される。揮発性メモリ3には、書込の時間的順序を維持するためのタイムスタンプ5、論理アドレスからディスク装置2へアクセスするための物理アドレスへの変換を行なうための変換マップ8が設けられている。また、不揮発性メモリ4には、ディスク装置2に書込むデータをログ構造化して保持するための書込みバッファ6として使用する領域、およびバッファ管理情報(バッファ管理テーブル)7を格納する領域が設けられている。バッファ管理情報7は、書込みバッファ6内に保持されている書込みデータの論理アドレスを管理するためのテーブル情報である。制御装置1は、タイムスタンプ5、書込みバッファ6、およびバッファ管理情報7を管理することにより、ディスク装置2に対するデータの入出力を制御する。

(書込みバッファ6とバッファ管理情報7との関係) 図2から図5は、不揮発性メモリ4に割り付けられた書込みバッファ6とバッファ管理情報7との関係を示す図である。

【0015】本実施形態のシステムでは、制御装置1は、外部のホストシステム10から書込み要求された書込みデータを、ディスク装置2に対して即書込み処理せずに、当該書込みデータをデータ長に応じて最適なブロック単位に分割し、書込みバッファ6にログ形式で順番に詰めて格納する。このとき、ホストシステム10からの書込みデータの論理アドレスを、書込みバッファ6の格納領域に対応するバッファ管理テーブル7のエントリに格納する。

【0016】バッファ管理テーブル7のエントリには、エントリの状態を示すフラグ(F, U, C)が設定される。フラグFは、エントリにデータが割り当てられた事を示す。フラグUは、エントリが使用されていない状態を示す。フラグCは次のエントリが連続したデータにより割り当てられている状態を示す。制御装置1は、バッファ管理テーブル7を参照することにより、ホストシステム10から受取った書込みデータを格納すべき次の書込みバッファ6の格納位置を決定できる。図2に示す具

体例では、書込みバッファ6のバッファ領域(格納位置)B0からB7まで書込みデータが格納されている。LA99、LA100、L35、L678…、LA541は、格納されている書込みデータの論理アドレスを意味する。

【0017】ここで、本実施形態では、書込みバッファ6からディスク装置2への書込み処理はストライプ単位であり、読み込み処理は可変長の論理ブロック単位であると想定する。ストライプ単位の書込み処理とは、N台のディスクドライブにおいて、 $N \times K$  (Kブロック数を示す整数)個の論理ブロックのデータの一括書込み動作の実行である。

(本実施形態の動作の説明) まず、図2を参照して、書込みバッファ6に格納された論理ブロックのデータの前の論理アドレスに位置する論理ブロックに、ホストシステム10から書込み要求があった場合の動作を説明する。本実施形態では、説明を簡略にするために、書込みバッファを書込みバッファと読み込みバッファとの共有バッファとして想定しているが、これらのバッファはそれぞれ独立であっても問題ない。この場合、当然ながら読み込みバッファは揮発性メモリまたは不揮発性メモリのいずれにあってもよい。

【0018】制御装置1は、ホストシステム10から書込み要求された書込みデータに対応する論理アドレスLA34の場合において、バッファ管理テーブル7を参照して、次に書込み可能な書込みバッファ6のバッファ領域B8(フラグUにより現時点では未格納領域)を認識する。このとき、書込み用の論理アドレスとして連続したバッファ領域を割り当てることが可能ように、書込みバッファ6上の格納位置(バッファ領域)を移動して、バッファ領域B2を格納位置として決定する。即ち、図3に示すように、制御装置1は、論理アドレスLA34のデータをバッファ領域B2に格納し、バッファ領域B8まで順次移動してデータを格納する。

【0019】次に、書込みバッファ6に格納された論理ブロックのデータ更新処理において、2論理ブロックにまたがる書込み要求の場合の動作を説明する。

【0020】ホストシステム10から書込み要求された書込みデータに対応する論理アドレスLA100の場合において、制御装置1は、図2に示すバッファ管理テーブル7を参照して、次の書込み可能なバッファ領域B8及び論理アドレスLA100に対応するバッファ領域B1を認識する。ここで、データ更新処理として、2論理ブロックにまたがるため、論理アドレスLA100のデータサイズが増えた分の論理アドレスLA101に対応するバッファ領域を割り当てる必要がある。この場合には、遅延したストライプ毎の書込み処理時に、ストライプ上の物理ブロックとして連続に割り当てるようにする。このときのブロック移動は最小で済むようにすると効率が良い。このデータ更新処理では、図4に示すよ

うな状態となる。

【0021】さらに、論理アドレスとして連続する2ブロックの読出し要求に対して、物理アドレスとして連続的に配置されていない場合に、物理アドレスとして連続化するための作を説明する。

【0022】ホストシステム10から要求された論理アドレスがLA111、LA112の場合において、制御装置1は、図2に示すバッファ管理テーブル7を参照して、次の書き込み可能なバッファ領域B8、及び論理アドレスとして連続したバッファ領域を割り当てることが可能であるバッファ領域B4、B5を決定する。ここでは、論理アドレスLA111及びLA112のそれぞれに対応するデータが、ディスク装置2上の別々のストライプに配置されている場合を想定するので、2回の読出し処理が必要となる。そこで、連続したブロックとしてまとめることにより、当該ストライプが書き込まれた後では、読出し処理のためのディスク装置2への読込み要求は、同一ストライプ上の連続した物理ブロックへの要求を一度だけ実行すればよい。この場合の書き込み処理では、図5に示すような状態となる。

【0023】以上のように書き込みバッファ6とバッファ管理テーブル7との関係について説明した。図2から図5に示す具体例では、書き込みバッファ6に対する読込み処理や書き込み処理時に連続した論理アドレスとして位置するデータに対して、物理的にも連続したアドレスになるようにデータの再配置が行なわれる。但し、コンピュータシステムのデータ転送の性能に基づいて、書き込みバッファ6の全てのデータを論理アドレスの順番に並び替える方法でもよい。また、データを並び替える代わりに、書き込みバッファ6からディスク装置2に書出すブロックの順を示したリストを備えることにより、データ移動を伴うことなく、リスト中の順番を変更する方法でもよい。

【0024】また、ディスク装置2（ディスクドライブ21～24）は、それぞれブロックサイズの整数倍（K）であるストライプユニットと呼ぶ予め決められた単位（ディスク上の1トラック長に近いサイズが良い）で書き込み処理を一括して実行する。このとき、ディスクドライブ21～24の物理的に同じ位置のストライプユニットは1つのストライプとして、同じタイミングで書き込み処理が行われる。また、ホストシステム10には、実際のディスクドライブ21～24を合わせて全記憶容量よりも少ない容量のディスク装置2として見られている。具体的には、ホストシステム10が初期時にディスク装置2の記憶容量を問い合わせて来ると、制御装置1は実際の記憶容量より少ない記憶容量を応答する。従って、ホストシステム10が論理的に書き込み、読出し可能な記憶領域以外に、ディスク記憶システムとしては独自の記憶領域を確保することが可能となる。この記憶領域を、本実施形態では空領域と呼ぶことにする。

【0025】さらに、タイムスタンプ5は、ホストシステム10からの書き込みデータが実際にディスク装置2に書込まれるときに付加される情報であり、ディスク装置2内でのデータ書き込み順序を判定するのに使用される。書き込みバッファ6のデータがディスク装置2に書込まれる毎に、タイムスタンプ5はインクリメントされることになる。

（システムの書き込み動作）以下図1のシステムにおいて、本実施形態の書き込み動作を主として図6を参照して説明する。

【0026】制御装置1は、ホストシステム10から書き込みデータとその論理アドレスを受取ると、不揮発性メモリ4上の書き込みバッファ6の空領域にブロック単位に分割して詰めて格納する。また、受取った論理アドレスはブロック毎のアドレスに変換して、バッファ管理テーブル7の対応するエントリに格納する。これらの処理は、受取った論理アドレスおよび書き込みバッファ6に既に格納されているデータを参照することにより、書き込みバッファ6の最適な格納位置に詰めて格納する。なお、既に書き込みバッファ6に格納されているデータに対するデータ更新処理の場合には、書き込みバッファ6の空領域に詰めて格納するのではなく、直接に書き込みバッファ6内の旧データを更新する。

【0027】ホストシステム10からの書き込みデータは、1ストライプ分に1ブロック少ない数（ $N \times K - 1$ ）だけ書き込みバッファ6に蓄積された段階で、制御装置1はそれらのデータをディスク装置2に書き込み処理する。このとき、最後の書き込みブロックとして、バッファ管理テーブル7に格納された各ブロックの論理アドレスと揮発性メモリ3上のタイムスタンプ5から論理アドレス・タグ・ブロックを作成する。この論理アドレス・タグ・ブロック内のアドレスデータとデータブロックとの間には、1対1の関係があらかじめ設定されており、各データブロックの論理アドレスが分かるようになっている。

【0028】この後に、制御装置1は、当該論理アドレス・タグ・ブロックを加えた1ストライプ分のデータを、一括してディスクドライブ21～24の空領域に同時に書き込む。前述したように、1ストライプ内の書き込みブロックは、論理アドレスに対して連続した領域になっているため、最新のデータは同じストライプ上に連続した配置となるため、読込み性能が向上する。このような動作を図6に示す。

【0029】ここで、タイムスタンプ5の値は書き込み処理が完了した段階でインクリメントされる。このように、多数の細かいディスク書き込み処理を1回で一括して実行できるため、ディスク書き込み性能を向上させることができる。

（データ更新処理時の詰替え処理）次に、図7を参照してデータ更新処理時の詰替え処理について説明する。

【0030】データ更新処理時では、旧データの領域を直接書き換えるのではなく、更新データを蓄積し、ディスク装置2内に予め用意した別の空領域に一括して書き込む方法では空領域が常に存在することが必須である。このため、ホストシステム10からのディスクアクセスが空いている間に、既に他の領域にデータが書込まれて無効になっているデータを寄せ集めて空領域を作る必要がある。この処理を詰替え処理と呼ぶ。この詰替え処理は、無効ブロック判定処理とストライプ統合の2つの処理からなる。

【0031】無効ブロック判定処理の具体例として、図7に示すように、ホストシステム10からのデータ書き込み要求において、1ブロックサイズの書き込み順序を想定する。図7において、L18などの「L××」はホストシステム10から渡される論理ブロック（論理アドレス）を意味し、S1などの「S××」は書き込み順番を表わす。本実施形態では、書き込みバッファ6は15論理ブロックのデータを保持できる。最初のS1～S15の書き込みデータが、1つのストライプ（ST1）にまとめられて、タイムスタンプT1が付加されてディスク装置2の空領域に書き出される。同様に、S16～S30の書き込みデータが別のストライプ（ST2）としてタイムスタンプT2が付加されて別の空領域に書き出される。なお、書き出し毎に、タイムスタンプ5はインクリメントされるので、「T1<T2」の関係がある。

【0032】ここで、図7から明らかなように、論理ブロックL9、L18のデータはタイムスタンプT1のストライプではS5、S2として、タイムスタンプT2のストライプではS19、S21のブロックとして重複して存在する。書込まれた順番を考えると、S19、S21のデータブロックが有効であり、S5、S2のデータは無効と判定されなければならない。しかし、ここでは便宜上、使用した書き込み順番S××は実際のディスク上には記録されていない。

【0033】そこで、ストライプ内の論理アドレス・タグを使用して、この無効ブロック判定処理を行なう。図7の具体例において、2つのストライプST1、ST2の論理アドレス・タグTG1、TG2の内容は、図8に示す通りである。図8から明らかなように、2つの論理アドレス・タグTG1、TG2には同じ論理ブロックL9、L18のデータが含まれている。ストライプST1のブロックB5、B2及びストライプST2のブロックB4、B6のいずれかのデータが無効である。また、論理アドレス・タグTG2には有効なデータである論理ブロックL18とL19が存在している。さらに、論理アドレス・タグTG1のタイムスタンプT1と論理アドレス・タグTG2のタイムスタンプT2とを比較すると、「T1<T2」の関係にあることから、ストライプST1のブロックB5、B2が無効であることが判定できる。以上のように、ディスク装置2内の論理アドレス・

タグを参照することにより、無効なデータブロックを見つけることができる。なお、このような処理に必要な読み書きに伴うバッファは書き込みバッファ7とは独立にした方が性能が向上する。当然ながら、書き込みバッファ7と共有してもよい。

【0034】ストライプ統合処理の具体例として、図9に示すように、2つのストライプST3、ST4を1つのストライプST5に統合する場合を想定する。図9に示すように、ストライプST3ではB2、B7、B8、B12、B13の5ブロックが有効であり、他の10ブロックは無効（ハッチング）であるとする。同様に、ストライプST4ではブロックB18、B19、B20、B21、B22、B24、B25、B27、B29の9ブロックが有効であり、他の6ブロックが無効（ハッチング）であるとする。2つのストライプの有効ブロックは合わせて14ブロックしかないので、この2つのブロックを1つに統合することにより、結果として1つの空領域が作れる。

【0035】ストライプ統合では図9に示すように、2つのストライプST3、ST4を揮発性メモリ3内に読みだし、有効なブロックだけを詰めて書き込みバッファ6に移す。この場合に、ストライプST1とストライプST2の中にある有効なブロックの中から論理アドレスが連続になるようにする。こうする事により、統合されたストライプST5のブロックは、論理アドレスとしてだけでなく物理アドレスとしても連続したデータを保持できることになる。これに合わせて、図10に示すように、論理アドレス・タグもTG3、TG4から有効ブロックの論理アドレスだけを対応する位置に移し、新しい論理アドレス・タグTG5を作成し、その時点のタイムスタンプ5を更新する。図10から明らかなように、新しく作成された論理アドレス・タグTG5に含まれる論理ブロック（アドレス）L13、L14とL22とL23は、ストライプST5上でも連続な領域となっている。すなわち、論理的にも物理的にも連続な領域が確保されていることになる。

【0036】この具体例のように、14個の有効ブロックを想定する場合には、さらにホストシステム10から1つの書き込みブロックを有するストライプを完成させて、ディスク装置2の空領域に一括して書き込むことになる。このとき、論理アドレスの並替えを行なっても問題はない。この場合、ディスク領域は有効に活用されるが、ホストシステム10からバーストでディスクアクセスがある場合には、書き込み処理を待たすために、ディスクアクセスが集中する可能性が高い。そこで、最後のデータブロックは空状態のままにして、アクセスが空いている間に書き込むことも可能である。このとき、論理アドレス・タグTG5の最後のデータブロックに対する論理アドレスには-1等NULLアドレスを入れることにより、データが入っていないことを表し、データが格

納されているデータブロックに対して論理アドレスの並び変えをしても問題は発生しない。

【0037】次に、書込まれたデータブロックの読出し動作について説明する。

【0038】前述の詰替え処理における無効ブロック判定処理を、ディスク装置2内の全ストライプの論理アドレス・タグに対して行うことにより、全論理アドレスに対する有効ブロックの物理的位置を検出できる。従って、原理的には、ホストシステム10から読出しブロックの論理アドレスを受け取る度に、全ストライプのチェックを行うことにより、読出すべき物理ブロックを見つけ出すことが出来る。しかし、この方法ではブロック読出し処理に膨大な時間を要するため実用的ではない。

【0039】そこで、システム起動時にだけ、全ストライプの論理アドレス・タグの調査を実行して、揮発性メモリ3上に論理アドレスから物理アドレスへの変換マップ8を作成する。ホストシステム10からの読出し要求に対しては、当該変換マップ8を使用して有効ブロックへのアクセスを実行する。これにより、常に論理アドレス・タグの調査をしなくても良く、読出し時に性能が大きく低下することはない。また、この変換マップ8は何時でも全ストライプを調査することで再生できるため、従来のように電源障害に備えて不揮発メモリ4上に格納する必要はない。

(変換マップの構成) 以下図11から図13を参照して、本実施形態の変換マップ8の構成について説明する。

【0040】変換マップ8は、図11に示すように、各論理アドレスに対するブロックが格納されているストライプ番号ST#、当該ストライプ内のブロック番号BLK#、さらにタイムスタンプTS#をテーブル形式で保持しているテーブル情報である。制御装置1は、論理アドレスL0～Lnにより変換マップ8を参照して、ST#とBLK#から実際の物理アドレスを求める。

【0041】ここで、システム起動時には、全ストライプの論理アドレス・タグを調査し、ディスク装置2内の全てのストライプに対して、無効ブロックを検出する。次に、全てのストライプに対してストライプ統合処理を実行する。この場合には、各論理アドレスに対応するブロックが格納されているストライプ番号ST#と、当該ストライプ内のブロック番号BLK#と、タイムスタンプTS#とから、論理アドレスの連続性を考慮して、ストライプ番号ST#、ブロック番号BLK#をできる限り連続的になるように再配置する。これにより、連続したデータの読込み時に、ストライプ番号ST#が異なったり、同じストライプ番号ST#でブロック番号BLK#が不連続になるという可能性が少なくなり、ブロック毎に行なっていた読込み要求は、書込みバッファ6とバッファ管理テーブル7との関係からも分かるように、複数ブロックを一度に読込むことが可能になり、性能低下

を軽減できる。

【0042】図11に示す変換マップの具体例では、ディスク装置2へ書込み処理または更新処理のためにアクセスする場合には、変換マップ8の全てのエントリを参照することになる。この方法では、全てのエントリを検索しなければならず、検索中に変換マップ8の全体を排他制御しなければならないといった制約があるため、効率的なアドレス変換であるとはいえない。そこで、効率的なアドレス変換を可能にするために、論理アドレスをハッシュキーにした変換マップ(図12を参照)や論理アドレスを2分岐検索する変換マップ(図13を参照)を作成することが望ましい。これらの変換マップを使用することにより、効率的なアドレス変換が可能となる。

【0043】論理アドレスをハッシュキーにした変換マップの場合には、頻繁に使用される論理アドレスのエントリをハッシュリストの先頭に配置するように作成する。このことにより、読み出したデータを更新するときに、更新性能が向上する。本構造で変換マップを作成する場合には、検索時間を考慮して、根の論理アドレスに考慮する必要がある。図13の具体例では、コンピュータシステムで取扱い可能な論理アドレスの中間値を使用した2分岐による本構造であるが、多分岐による本構造も可能である。

【0044】また、システム起動時の変換マップの作成は、調査した論理アドレス・タグの全論理アドレスについて、テーブルのタイムスタンプ5より論理アドレス・タグのタイムスタンプ5が大きいときだけ、そのストライプ番号ST#と対応するブロック番号BLK#を各々の変換マップに応じてテーブル、ハッシュリスト、本構造に登録する。この調査を全ストライプについて行なえば、有効ブロックだけを指す変換マップが作成される。この場合にも、ストライプ統合処理と同様の処理をすることにより、連続したデータの読込み処理時に、ストライプ番号ST#が異なったり、同じストライプ番号ST#でブロック番号BLK#が不連続となる可能性が少なくなり、性能を低下することがなくなる。更に、ディスク装置2にストライプを書込む毎に、その論理アドレス・タグに対して同様の処理を行なうことにより、この変換マップは常に有効なブロックを指す。また、ディスクアクセスが空いているときに、各ストライプの論理アドレス・タグと変換マップとを比較検査することにより、離散しているデータを同一のストライプや連続したブロックに統合することができる。更に、メモリ障害等でこの変換マップが不正な値になっても検出可能である。

【0045】以上のような変換マップ8は、ディスク装置2の記憶容量に応じて、書き込み処理の時間的順序を維持するために必要なタイムスタンプ5と、論理アドレスから物理アドレスへのアドレス変換を行なうため揮発性メモリ3の記憶容量とが必要となる。このため、記憶容量が大きなディスク装置2の場合には、メインメモリ



上の揮発性メモリ3が多く必要となるため実際的ではない。そこで、最新データの位置情報を管理する間接マップの一部を、ディスク装置内の空領域に格納し、変換するデータとその論理アドレスとが間接マップに存在しない場合には、保存してあるディスク装置2内の空領域から間接マップを読みだし、間接マップを再作成することが有効になる。

【0046】ここで、間接マップを再作成する場合について説明する。論理アドレスから物理アドレスに変換する度に、ディスク装置に保存してある間接マップを読み出して再作成する方式では、コンピュータシステムの性能上から容認しがたいものになってくる。そこで、ディスク装置内に保存してある間接マップの一部に対して、高頻度の読出し要求が発生しないようにすることが重要となる。そのためには、各論理アドレスに対応する間接マップのエントリのタイムスタンプの古いものから順番にディスク装置2へ保存することにより、最適な間接マップを構成することが可能である。

【0047】以上のように、変換マップ8の作成処理における要素は論理アドレス・タグの検査である。故に、大容量のディスク装置2のように論理アドレス・タグ数が多い場合、電源障害やシステム起動時の変換マップ作成に長時間を要する。特に、図2に示すように、論理アドレス・タグ・ブロックが1台のディスクドライブ24に集中すると、システム起動時には当該ディスクドライブ24にアクセスが集中し、論理アドレス・タグの調査を並列に行うことが不可能となる。そこで、図14に示すように、ストライプにより論理アドレス・タグが格納されるディスクドライブを4台に分散して並列に論理アドレスタグを調査することにより、この変換マップ作成に要する時間を1/4に短縮化できる。

(セグメント分割管理) 図15と図16はディスク装置2の記憶領域を複数のセグメントに分割管理するセグメント分割管理方式を示す図である。この方式により、変換マップ8の作成に必要な論理アドレス・タグの検査数を削減することができる。図16にセグメント分割方式におけるディスク装置の記憶領域の構成を示す。図16に示すように、ディスク装置の記憶領域は、ストライプを単位としてセグメント管理情報(ハッチング)と4つのセグメントに分割される。ここで、セグメントとは書込みバッファデータの一括書込み処理や詰め替え処理のディスク書込みがある期間集中して行われる単位領域のことである。例えば、セグメント2がディスク書込みの対象である間は、セグメント1、3、4には書込み処理が実行されないように空領域の選択を制御する。

【0048】また、あるセグメントの空領域が少なくなりディスク書込みを他のセグメントへ切替えるときにはセグメント管理情報をディスク装置に保存する。セグメント管理情報は図15に示すように、セグメント番号と切替え時変換マップから構成される。セグメント番号と

は切替え先のセグメント番号で、切替え時変換マップとはセグメントを切替える時点での揮発性メモリ3上の変換マップ8の状態を示す。

【0049】なお、切替え時変換マップはセグメントが切替える度に全て上書きするのではなく、直前のセグメントに書込まれた論理アドレスのエントリだけを書き戻せばよい。従って、前回のセグメント切替え時にタイムスタンプを覚えておき、変換マップのタイムスタンプを比較することにより、直前のセグメントに書込まれた論理アドレスを判定できる。

【0050】このセグメント分割方式では、セグメント切替え時にセグメント管理情報を保存している。よって、セグメント切替え時の変換マップをセグメント管理情報から読み出して、その後でセグメント管理情報のセグメント番号で指示されるセグメントの論理アドレスタグだけを検査するだけで、全論理アドレス・タグを検査した場合と同じ変換マップが再現できる。従って、この方式により必要な論理アドレス・タグの検査数は1セグメント分で良く、この例では変換マップの作成に要する時間を1/4に短縮化することが可能となる。

【0051】更に、不揮発性メモリ4上にセグメント内の全ストライプに対応したビットマップを用意して、セグメント切替え時にはこのビットマップをクリアし、一括書込みや詰め替えの書込み時には書込んだストライプに対応するビットを“1”にセットする。これによりセグメントを切替えてから変更のあるストライプだけがビットマップが“1”になる。従って、変換マップ作成時にこのビットマップを参照し、変更の有ったストライプの論理アドレスタグだけを検査することで検査数をさらに減らせ、変換マップ作成に要する時間を更に短縮化できる。

【0052】通常、論理アドレス・タグのサイズは512～1024バイトであり、ディスクのシーケンシャルアクセスとランダムアクセスに約50倍の性能差がある。図2に示す方式では論理アドレス・タグの情報が各ストライプ毎にとびとびに存在するので、変換マップの作成では時間のかかるランダムアクセスを実行している。そこで、図17に示すように、論理アドレス・タグだけを連続して格納する専用タグ領域を(セグメント分割する場合は、各セグメント毎に)用意し、50倍も高速なシーケンシャルアクセスで論理アドレスタグを読み出せるようにする。そして、ホストシステム10からのデータの一括書込み処理や詰め替え処理時には、空領域だけでなく対応する専用タグ領域にも論理アドレス・タグを書込むようにする。この方式であれば、1ストライプ当たり4回のディスク書込み処理に対して、専用タグ領域への論理アドレス・タグの書込み処理のために、書込み操作が1回増えることになる。しかし、変換マップ作成が50倍も高速になるので、ディスク装置の立ち上がり時間が問題となるとときには非常に有効な手段である。

専用タグ領域への書込み時間を最小にするためには、専用タグ領域は図17に示すように、対象領域を中心に設定することにより、ディスクドライブのシーク時間を短縮化することが望ましい。また、通常ではディスク装置2はセクタ(512バイトなど)単位の書込み処理を実行するため、専用タグ領域内の論理アドレスタグはセクタ単位で割り付けて、論理アドレス・タグの書込み処理時には読出し処理を不要にすることが望ましい。

(タイムスタンプの説明) 図1に示すように、タイムスタンプ5は揮発メモリ3上に記憶されている。このため、システムの電源障害などにより揮発メモリ3上のタイムスタンプ5が消失する可能性がある。そこで、変換マップ8の場合と同様にして、システム起動時にだけ全ストライプの論理アドレス・タグを調査し、一番大きなタイムスタンプ5の次の値を揮発メモリ3上のタイムスタンプ5にセットする。なお、変換マップの作成処理の説明で述べた時間短縮手法がそのままタイムスタンプの再生にも適用できる。

【0053】また、タイムスタンプ5はディスク装置2に書込む度に、インクリメントされて、ディスク上の書込み順序の判定処理のみに使用される。具体例として、タイムスタンプ5が24ビットのカウントで構成される場合を想定する。24ビットカウントでは、16M回の書込み処理でカウンタが一周してクリアされてしまう。そこで、一般的には有効なタイムスタンプ5の最小値を基準として、それより小さい値は16Mを加えて比較して判定する。この最小値も同様にシステム起動時にだけ全ストライプの論理アドレス・タグを調査して求める。しかし、この手法が使えるのはタイムスタンプの最大値が最小値を追い越さないこと、つまり、タイムスタンプの最大値と最小値との差が24ビットで表わせる範囲以内であることを前提にしている。従って、タイムスタンプ5が一周前に必ず全ストライプを更新してタイムスタンプ値を新しく更新する必要がある。これには、無効ブロックが少なくとも予め設定した書込み回数に更新されなかったストライプを詰替えの対象として選ぶように制御するか、無効ブロックの論理アドレスをNULLアドレスにした、そのストライプの論理アドレスタグだけを書き換える。NULLアドレスを使う方法は論理アドレス・タグ・ブロックの書き換えであるので詰替え処理と比較して常に軽い処理である。

【0054】尚、前記の具体例では、無効ブロック判定処理に2つストライプST1、ST2の論理アドレス・タグを相互に比較して判定する方法についてのみ説明したが、全無効ブロックを調べるには2つのストライプ間の全組み合わせを調べなければならない。しかし、変換マップがあれば論理アドレス・タグ内の各論理アドレスについて、有効データを指示している変換マップのタイムスタンプとそのストライプのタイムスタンプとを比較し、値が小さいブロックを無効ブロックと判定すること

ができる。

(本実施形態の変形例) 図18は本実施形態の変形例を示す図である。本実施形態は、RAID0方式を適用したシステムを想定している。これに対して、本変形例はパリティデータを使用する冗長性ディスク構成のRAID5方式を適用したシステムである。

【0055】本システムは、図18に示すように、図1のシステムに対して、冗長用のディスクドライブ25が追加された構成である。なお、これ以外の制御装置1、ディスク装置2(ディスクドライブ21、22、23、24)、揮発性メモリ3、不揮発性メモリ4、タイムスタンプ5、書込みバッファ6、およびバッファ管理情報(テーブル)7は、図1に示す本実施形態と同じ機能を有する要素である。

【0056】以下、本変形例のシステムの動作を、本実施形態との差異を中心にして説明する。書込み処理では、ホストシステム10からの書込みデータが1ストライプ分に1ブロック少ない数( $N \times K - 1$ )だけ書込みバッファ6に蓄積された段階で、制御装置1はそれらのデータをディスクドライブ21~25に書込み処理していく。このとき、最後の書込みブロックとして、バッファ管理テーブル7に格納された各ブロックの論理アドレスと揮発性メモリ3上のタイムスタンプ5とから論理アドレス・タグ・ブロックを作成するまでは処理は、本実施形態の場合と同様である。

【0057】この後、制御装置1は、図19に示すように、論理アドレス・タグ・ブロックを加えた1ストライプ分のデータからストライプユニット毎の排他論理輪演算(XOR)を実行し、パリティデータ(P)のストライプユニットを作成する。そして、このパリティデータ付きのストライプのデータを一括して、ディスクドライブ21~25の空領域に同時に書込む。本実施形態のように、可変長のブロック単位での読込み処理を行なう場合には、書込むための空領域は、各ディスクの現在のストライプ位置からの差分が最も少なくなる空領域に近いストライプを選択する。このような選択方法により、各ディスクのシーク時間による書込み処理の応答時間を均等にすることができる。ストライプ単位での読込みによる場合には、このようなことにはならないので問題がない。

【0058】また、タイムスタンプ5の値は書込み処理が完了した段階でインクリメントされる。このように多数の細かいディスク書込み処理を1回にまとめられるのに加え、パリティ計算に必要な旧データや旧パリティデータのブロックを読込む必要がないので、さらにディスクアクセス回数を減少することができる。なお、ストライプ詰替え処理時のディスク書込み処理でも、同様にパリティデータ付きのストライプを作成してからディスク装置2に書込む(図19を参照)。

【0059】冗長性ディスク構成(パリティ)のRAI

D5方式のシステムでは、1台のディスクドライブが故障しても、故障したディスクドライブのデータはストライプを構成する他ディスクドライブのデータとパリティデータとのXOR演算を実行することにより再現することができる。しかし、システム起動時に一台故障していた場合には、論理アドレス・タグを格納していないディスクドライブのデータも読出して論理アドレス・タグを再生してから検査するため、変換マップ8の作成処理にかなり時間を要し、システム起動が完了するまでの時間が大幅に増大してしまう。

【0060】そこで、図20に示すように、ストライプを構成するデータブロックを1つ減らして2つのディスクドライブに同じ論理アドレス・タグを書込むように制御する。これにより、ディスクドライブの1台が故障しても、変換マップ8の作成時には残っている方の論理アドレス・タグを読出すことができるので、システム起動に要する時間の大幅な増大を回避できる。

【0061】また、変換マップ作成の高速化のために専用タグ領域を活用する場合、図21に示すように、論理アドレス・タグが専用タグ領域に格納されるディスクドライブと、ストライプに格納されるディスクドライブとが異なるように、専用タグ領域の論理アドレス・タグの割り付け処理を制御することにより、ストライプ内の論理アドレス・タグは1つでよい。

【0062】なお、専用タグ領域へ論理アドレス・タグを書込む場合も、パリティデータによるディスク障害対策を行うと、従来1回の書込み操作の増大で済んでいたものが、2回の書込み処理と2回の読出し処理が必要になって、一括書込み処理時やストライプ詰替え処理時のディスク書込み動作のオーバヘッドが大きく増大する。従って、この専用タグ領域の情報はパリティデータを使用した障害対策を実行しない方が望ましい。専用タグ領域の情報は変換マップ高速化のためであり、故障したディスクドライブの専用タグ領域に格納されていた論理アドレス・タグは、ストライプ中のものを（ランダムアクセスで）見れば良いので問題はない。また、ランダムアクセスで検査する論理アドレス・タグは1/5だけであるので、変換マップ作成の高速化の効果はある。

【0063】なお、本実施形態は、RAID方式のディスク記憶システムに適用した場合について説明したが、これに限る事なく、光磁気ディスク等を使用したシステムにも適用することができる。また、シーケンシャル書込み処理とランダム書込み処理とで大きく性能が異なるディスク記憶システムや、小ブロックの更新処理では2回の読出し処理と2回の書込み処理とが必要なパリティによる冗長性を持たせたRAID構成のディスク記憶システムにも適用できる。

【0064】

【発明の効果】以上詳述したように本発明によれば、ディスク装置から読出すべきデータを不揮発性メモリ上の

書込みバッファの空領域に物理的に連続したブロック単位に分割して詰めて格納する方式であるため、一括したアドレスに変換できるため、読出しのための入出力要求が分割される必要がない。また、ディスク装置の容量に依存しないでメインメモリ上に変換マップを作成することができるため、メインメモリの記憶領域を効率的に使用することができる。本発明を特に高速の書込み方法を採用したRAID方式のディスク記憶システムに適用すれば、特にデータ更新処理に関する性能の向上を図ることが可能となる。

【図面の簡単な説明】

【図1】本発明の実施形態のRAID方式のディスク記憶システムの概念的構成を示すブロック図。

【図2】本実施形態の書込みバッファとバッファ管理情報との関係を示す概念図。

【図3】本実施形態の書込みバッファとバッファ管理情報との関係を示す概念図。

【図4】本実施形態の書込みバッファとバッファ管理情報との関係を示す概念図。

【図5】本実施形態の書込みバッファとバッファ管理情報との関係を示す概念図。

【図6】本実施形態の書込み動作におけるディスク装置の空領域の格納内容を説明するための図。

【図7】本実施形態の書込み動作における詰替え処理を説明するための図。

【図8】図7の具体例におけるストライプST1、ST2の論理アドレス・タグTG1/TG2の内容を示す図。

【図9】図7の具体例におけるストライプ統合処理を説明するための図。

【図10】図9のストライプ統合処理において、論理アドレスタグTG3/TG4から論理アドレスタグTG5を作成する場合の具体例を示す図。

【図11】本実施形態の変換マップの構成を説明するための図。

【図12】本実施形態の変換マップの構成を説明するための図。

【図13】本実施形態の変換マップの構成を説明するための図。

【図14】本実施形態に係る論理アドレス・タグを分散配置した具体例を説明するための図。

【図15】本実施形態に係るセグメント分割管理を説明するための図。

【図16】本実施形態に係るセグメント分割管理を説明するための図。

【図17】本実施形態に係る専用タグ領域の内容を示す図。

【図18】本実施形態の変形例に係るRAID5方式のディスク記憶システムの概念的構成を示すブロック図。

【図19】本変形例の動作を説明するための概念図。

【図20】本変形例に関する論理アドレス・タグの作成方法を示す図。

【図21】本変形例に関する専用タグ領域を割付け方を説明するための図。

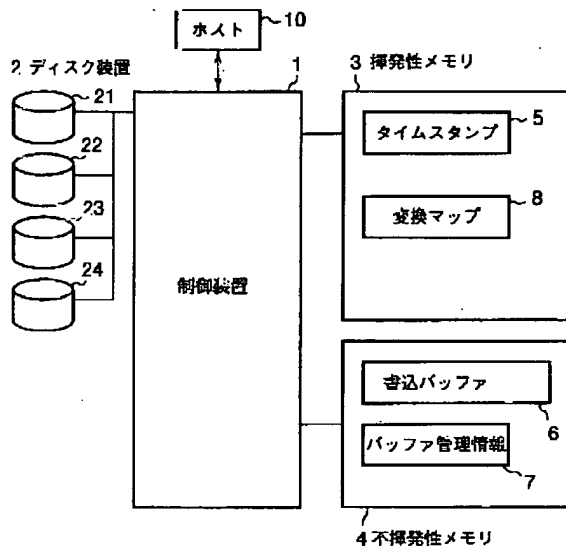
【図22】従来のディスク記憶システムのデータ更新方法を説明するための概念図。

【図23】本発明の実施形態のRAID方式のディスク記憶システムの概念的構成を示すブロック図。

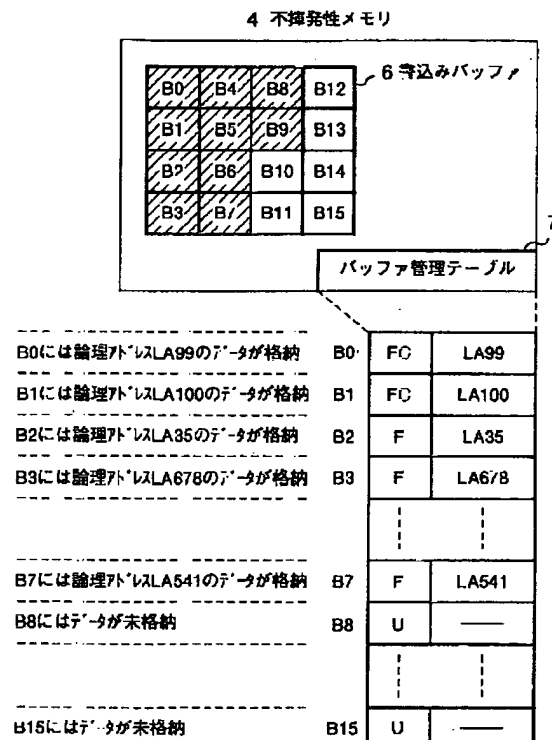
【符号の説明】

- 1…制御装置(CPU)
- 2…RAID方式のディスク装置
- 3…揮発性メモリ(メインメモリ)
- 4…不揮発性メモリ
- 5…タイムスタンプ
- 6…書き込みバッファ
- 7…バッファ管理情報(テーブル)
- 10…ホストシステム
- 21～25…ディスクドライブ

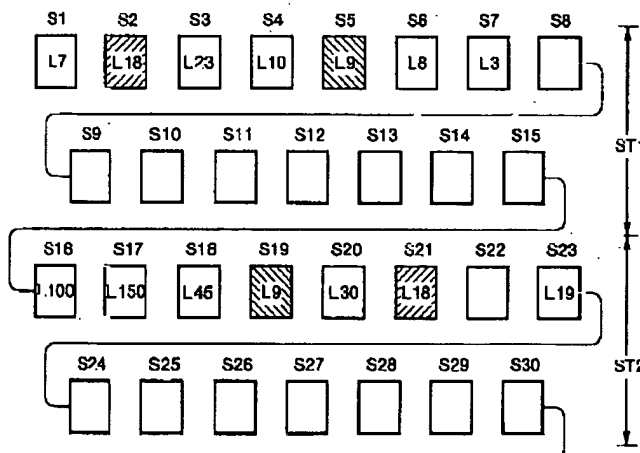
【図1】



【図2】



【図7】



【図11】

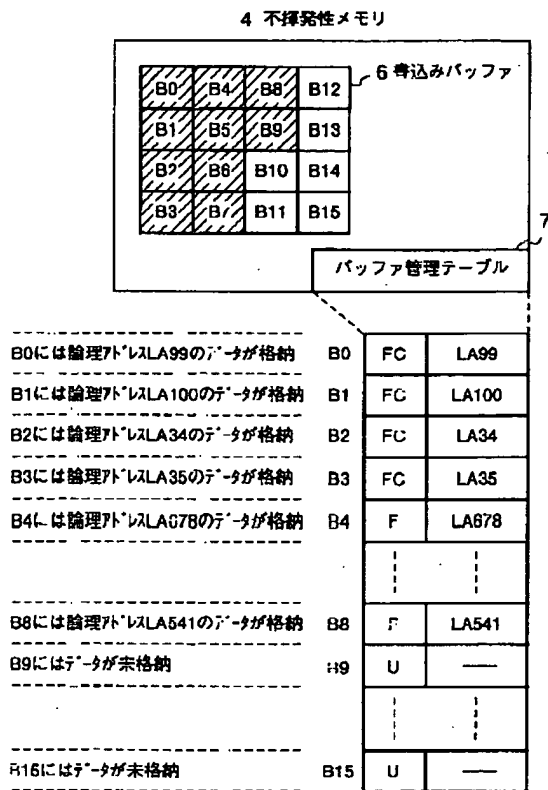
論理アドレス	ST#	BLK#	TS#
1.0			
1.1			
1.2			

【図15】

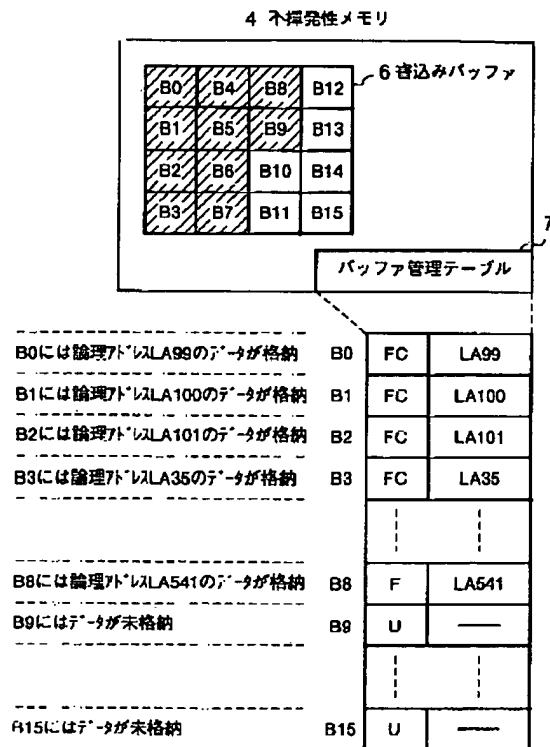
SG#	切替え時交換マップ
-----	-----------

セグメント管理情報

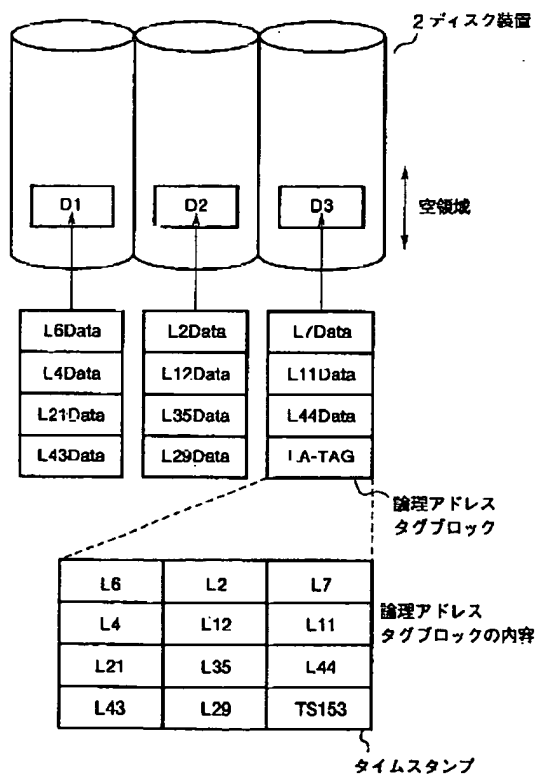
【図3】



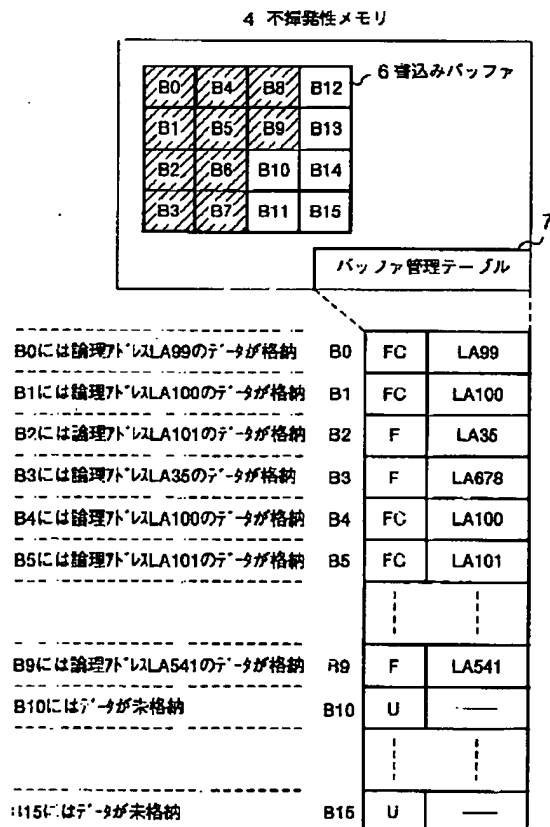
【図4】



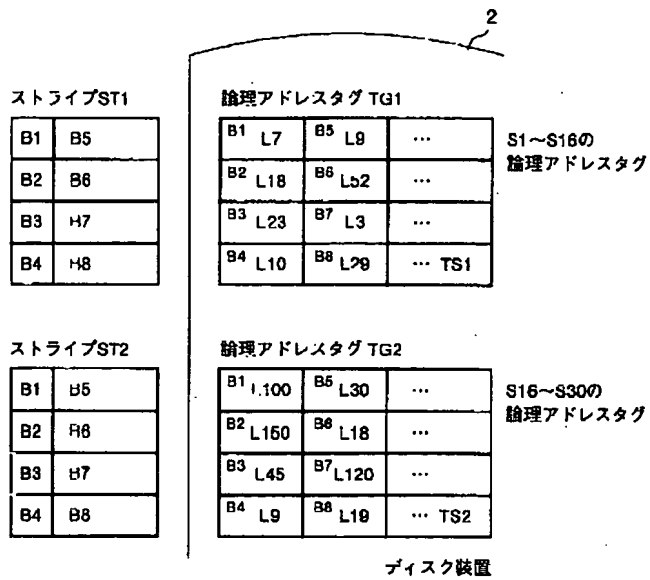
【図6】



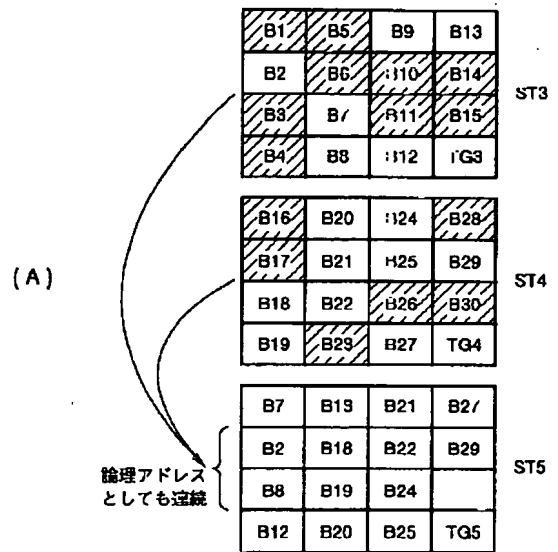
【図5】



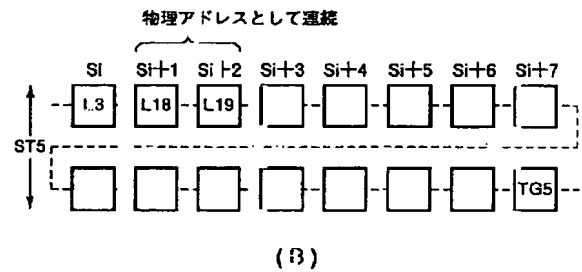
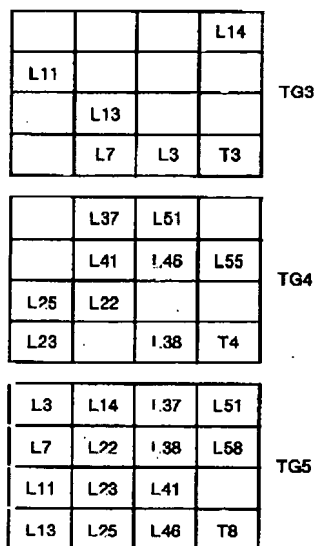
【図8】



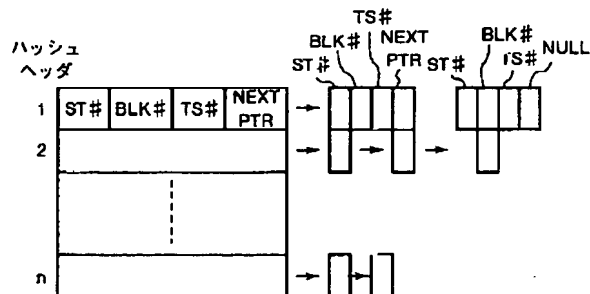
【図9】



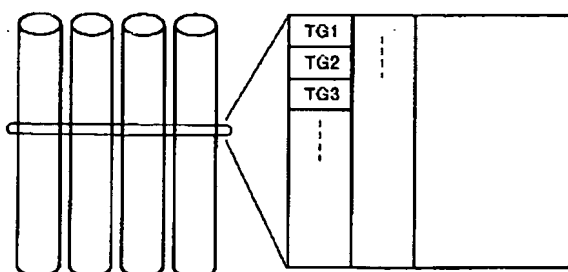
【図10】



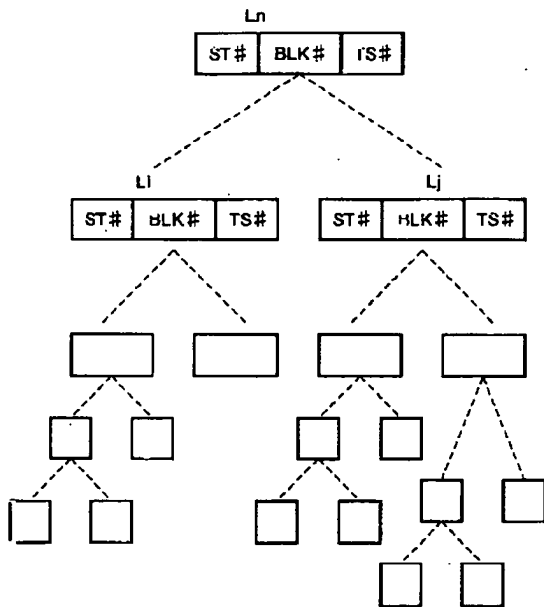
【図12】



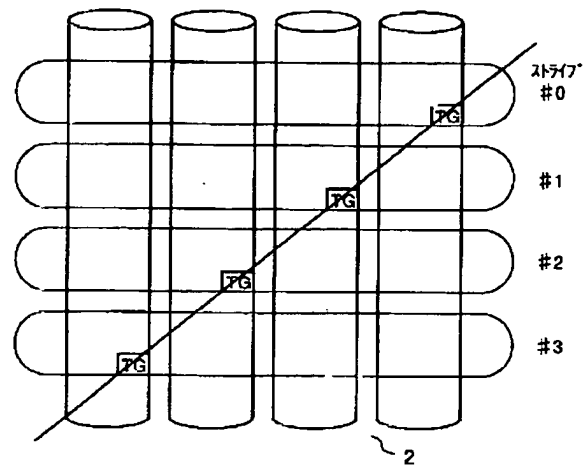
【図17】



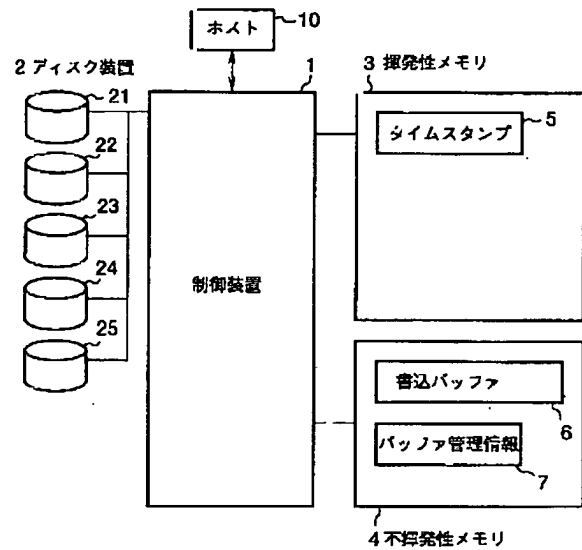
【図13】



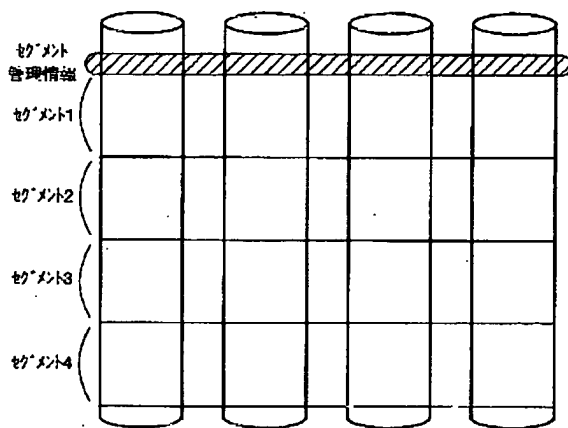
【図14】



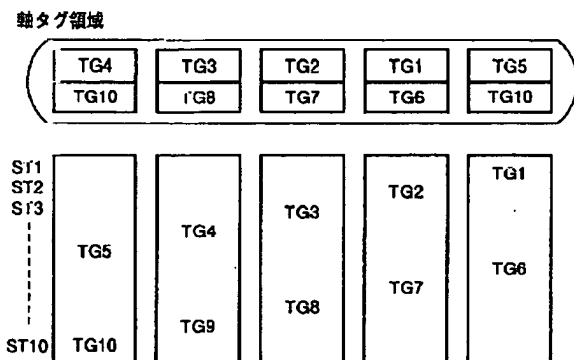
【図18】



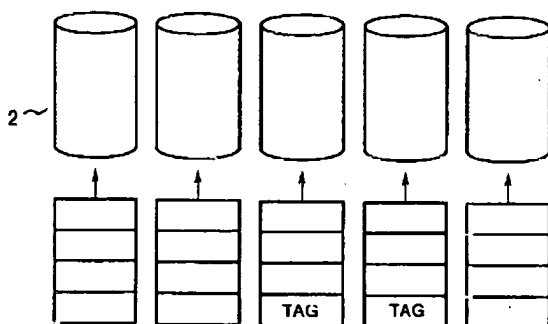
【図16】



【図21】

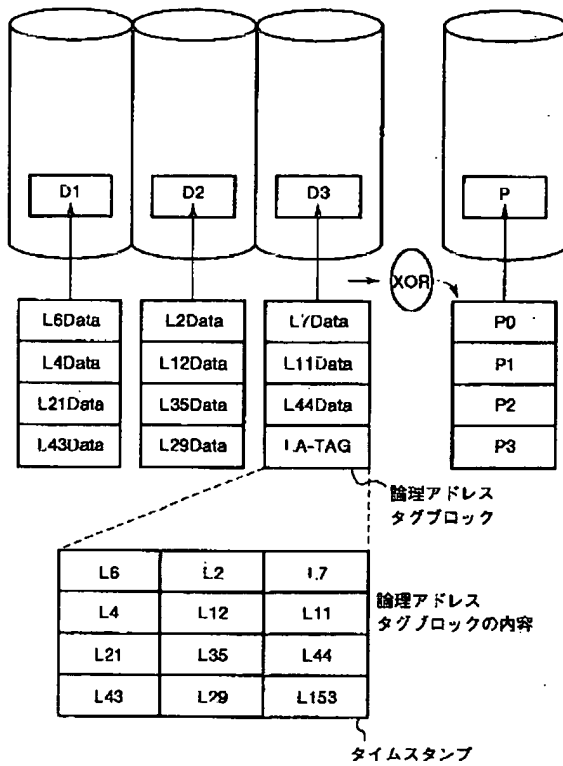


【図20】

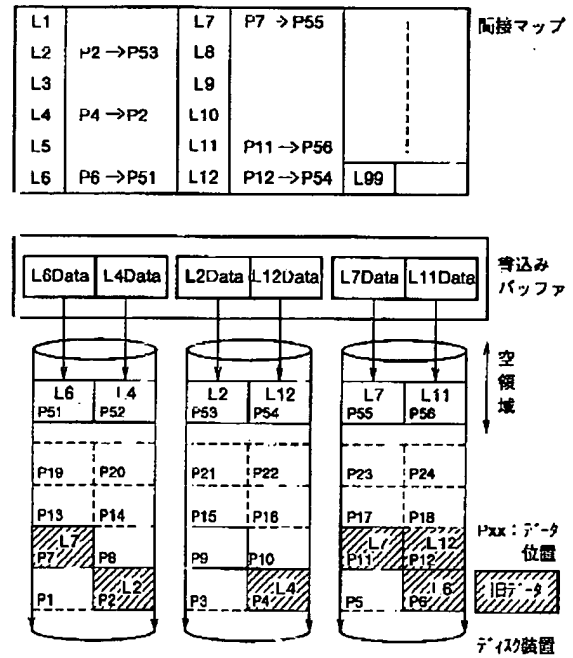


【図19】

2 ディスク装置



【図22】



【図23】

